Attorney Docket No. GP-054-00-US

METHODS AND APPARATUS FOR DETERMINING EQUIVALENT DESCRIPTIONS FOR AN INFORMATION NEED

JEFFREY DEAN, GEORGES HARIK,
BENEDICT GOMES, and NOAM SHAZEER

Assignee: **GOOGLE, INC.**
2400 Bayshore Parkway
Mountain View, California 94043
650-330-0100
a Corporation of the State of California

Status: Large Entity

Express Mail No. EL773547318US

METHODS AND APPARATUS FOR DETERMINING EQUIVALENT DESCRIPTIONS FOR AN INFORMATION NEED

BACKGROUND OF THE INVENTION

A. Field of the Invention

The present invention relates generally to information search and retrieval and, more particularly, to determining equivalent descriptions for an information need based on multiple references to that same information need.

B. Description of Related Art

The World Wide Web ("web") contains a vast amount of information. Locating a desired portion of the information, however, can be challenging. Unless the user is aware of the specific location of the desired information, the user must rely on a service to assist in locating the information. Typically, the user will identify the information sought via a query of some form, and the service will attempt to direct the user to the information based on the query.

Unfortunately, however, the user cannot always formulate the query in a sufficient manner as to obtain all of the information that the user desires. For example, the user may have an information need that can be described in multiple ways, but the user may only be aware of a limited way of describing that information need. In such a case, the user may obtain only a subset of the desired information.

It would be helpful, therefore, to have methods and apparatus for determining equivalent ways of describing an information need.

SUMMARY OF THE INVENTION

Systems and methods consistent with the present invention address this and other needs by determining equivalent descriptions for an information need based on multiple references to that same information need.

In one implementation consistent with the present invention, a method for determining equivalent descriptions for an information need involves identifying a list of queries issued by one or more users. A candidate pair of equivalent descriptions is identified by locating two queries that refer to the same information need. A score is calculated for the candidate pair, depending on the frequency with which the candidate pair occurs in the list. If the score exceeds a defined threshold, each half of the candidate pair is determined to be an equivalent description for the information need.

In another implementation consistent with the present invention, a method for determining synonyms includes obtaining a list of search queries issued by one or more users. The list is sorted first by user and second by the time when the query was issued. A set of adjacent queries for a single user is selected and, from this set, two queries are identified that contain at least one query term in common. A candidate synonym pair is identified based on the uncommon portions of the two queries, and a score is calculated for it based on the frequency with which the candidate synonym pair occurs in the list. If the score

exceeds a defined threshold, each half of the candidate synonym pair is determined to be a synonym of the other half.

Additional aspects of the present invention are directed to computer systems and to computer-readable media having features relating to the foregoing aspects.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, explain the invention. In the drawings,

Fig. 1 is a diagram illustrating an environment in which the present invention may be implemented;

Fig. 2 is a diagram illustrating a search engine consistent with the present invention;

Fig. 3 is a diagram illustrating an architecture in which the present invention may be implemented;

Fig. 4 is a flow diagram for identifying equivalent descriptions for an information need, consistent with the present invention;

Fig. 5 is a flow diagram for identifying synonyms based on a search query log, consistent with the present invention;

Fig. 6 is a sample query log consistent with the present invention;

Fig. 7 is a flow diagram for identifying equivalent descriptions based on anchor text information, consistent with the present invention; and

Fig. 8 is an illustrative hyperlinked document system.

DETAILED DESCRIPTION

The following detailed description of the invention refers to the accompanying drawings. The detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims and equivalents.

The present invention analyzes collections of descriptions to identify those that relate to the same information need. In one implementation, these descriptions are in the form of search queries issued to a search engine, which are then organized by user and by date. Queries within a certain adjacency level are analyzed to determine if they contain one or more terms in common. If they do, the uncommon terms are considered a candidate pair of equivalent descriptions for the same information need. Scores are calculated for the candidate pairs based on the frequency with which they appear in the collection, and those above a certain threshold are determined to be equivalent. Those skilled in the art will recognize that many other implementations are possible, consistent with the present invention.

A. Environment and Architecture

Fig. 1 is a diagram illustrating an environment in which the present invention may be implemented. The environment includes a requester 110, an information location tool 120, and information set 130.

Information set 130 represents the collection of information available for access. This information set 130 may be, for example, hypertext pages available over the Internet, or any other collection of documents or other information.

Requestor 110 represents an entity that is seeking to locate a subset of information set 130. Because the collection of information set 130 may be vast, requestor 110 may employ the aid of an information location tool 120.

Information location tool 120 facilitates finding information within information set 130. As described in more detail below in reference to Fig. 2, information location tool 120 analyzes information set 130 to develop an index. Information location tool 120 receives a request for information from requestor 110, compares the request to its index, and provides requestor 110 with pointers to the relevant portions of information set 130.

Fig. 2 is a diagram illustrating an information location tool 120 consistent with the present invention. In this implementation, information location tool 120 is a search engine 205. Search engine 205 may include a crawl component 210, a content repository 220, a content input component 225, an index 230, a query processing component 240, a relevancy component 250, an ordering component 260, and an output component 270.

Crawl component 210 analyzes and collects information from information set 130 and places the information into content repository 220. In an implementation involving the world wide web, for example, crawl component 210 locates web pages, collects them, and stores them in content repository 220. In addition to information stored by crawl component 210, the content repository

220 may also include information manually entered into (or otherwise located by) content input component 225.

Index 230 represents a distillation of the information stored in content repository 220. Much like an index to a book allows a user to locate information within the book much faster than analyzing each page of the book, index 230 facilitates location of desired information from information set 130. Index 230 may be implemented as a series of associations of (1) terms (e.g., words) and (2) the documents within which those terms appear. In addition to the documents associated with a particular term, index 230 may also contain information such as the location of the term within a given document, the number of times the term appeared in a given document, etc.

Query processing component 240 receives information requests from a requestor 110. Query processing component 240 may analyze the information requests to understand better what information the requestor 110 seeks. Particularly salient to the present invention, query processing component 240 may determine equivalent descriptions for the information need sought by requestor 110. Query processing component 240 may also perform functions such as detecting and correcting misspellings, transform the information request into a format that is more likely to produce the desired information, when processed by the remainder of search engine 205, etc.

Relevancy component 250 receives the (perhaps modified) information request from query processing component 240, and determines the items within information set 130 that are relevant to the information request. This may be

accomplished, for example, by comparing the terms of the information request with index 230.

Ordering component 260 receives a list of relevant items from relevancy component 250 and determines the order in which they should be presented. Typically, the relevant items will be ordered in a manner that maximizes the likelihood that the items most likely to be of interest to the user appear first. One example of an ordering system is described in S. Brin, L. Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine, WWW7 / Computer Networks 30(1-7): 107-117 (1998).

Output component 270 receives an ordered list of items from ordering component 260, formats that list into a manner suitable for presenting to requestor 110, and provides the ordered list to requestor 110.

Fig. 3 is a diagram illustrating an architecture in which the present invention may be implemented. The architecture includes multiple client devices 302, a server device 310, and a network 301, which may be, for example, the Internet. Client devices 302 each include a computer-readable medium 309, such as random access memory, coupled to a processor 308. Processor 308 executes program instructions stored in memory 309. Client devices 302 may also include a number of additional external or internal devices, such as, without limitation, a mouse, a CD-ROM, a keyboard, and a display.

Through client devices 302, requestors 110 can communicate over network 301 with each other and with other systems and devices coupled to network 301, such as server device 310.

Similar to client devices 302, server device 310 may include a processor 311 coupled to a computer readable memory 312. Server device 310 may additionally include a secondary storage element, such as database 330.

Client processors 308 and server processor 311 can be any of a number of well known computer processors, such as processors from Intel Corporation, of Santa Clara, California. In general, client device 302 may be any type of computing platform connected to a network and that interacts with application programs, such as a digital assistant or a "smart" cellular telephone or pager. Server 310, although depicted as a single computer system, may be implemented as a network of computer processors.

Memory 312 may contain a number of programs, such as the components described above in reference to Fig. 2.

B. Operation

Fig. 4 is a flow diagram for identifying equivalent descriptions for an information need, consistent with the present invention. The process begins by identifying a plurality of descriptions that are associated with a plurality of information needs, which collection of descriptions will hereafter be referred to as the "Data Set". (Stage 410). The plurality of descriptions may be in the form of search requests, as explained below in reference to Fig. 5, or a variety of other forms. For purpose of the present invention, all that is required is that there be descriptions of some form, and that each description be associated with one or more information needs.

Next, candidate pairs of equivalent descriptions are identified. (Stage 420). This may be accomplished by analyzing descriptions that are related in some manner. For example, if two descriptions contain a common term, one might deduce that they are related in some manner. In particular, one might deduce that the two descriptions are equivalent, or that the terms that are not in common are equivalent. Similarly, if two descriptions explicitly point to the same piece of information, one might deduce they are related in some manner.

A score is then calculated for each candidate pair. (Stage 430). The score may be calculated in an absolute manner (e.g., by determining the frequency with which the candidate pair occurs in the Data Set), relative to other candidate pairs (e.g., by determining how frequently the candidate pair occurs relative to other candidate pairs), or in a variety of other manners.

The scores may then be compared against some threshold to determine whether the candidate pair is an equivalent description for the information need. (Stage 440). The threshold may be set depending on the confidence required in the outcome.

Fig. 5 is a flow diagram for identifying synonyms based on a search query log, consistent with the present invention. In this implementation, the process begins by obtaining a list of search queries--e.g., a query log. (Stage 510). The query log may be maintained, for example, in the query processing component 240 of search engine 205.

An exemplary portion of a hypothetical query log is shown in Fig. 6. In a preferred implementation, the query log contains, for each query, information

about the user who submitted the query (i.e., a UserID), when the query was submitted (i.e., date and time), and the query itself. In addition to the foregoing, the query log may also include a list of information that was provided to the user in response, a record of any action taken by the user on the search results (e.g., whether the user clicked on any of the results), as well as other data concerning the query and user behavior.

The query log may then be sorted by user and by time. (Stage 520). For example, Fig. 6 shows queries submitted by three different users, represented as UserIDs 2, 1, and 3. As shown in Fig. 6, these queries are grouped by the UserID and are further ordered based on when they were submitted by the user.

Next, a set of adjacent queries are selected. (Stage 530). The scope of this selection can be limited to a particular user, can span a certain time frame, or limited in any other manner. For exemplary purposes, we will assume that the scope is limited by selecting a set of two consecutive queries issued by the same user. Under this assumption, the adjacent queries are as follows: 1 and 2; 2 and 3; 3 and 4; 5 and 6; 7 and 8; and 8 and 9. In practice, a window of two or five consecutive queries by the same user has been found to work well.

The selected queries are analyzed to determine if they contain one or more terms in common. (Stage 540). Using the example log in Fig. 6, it may be determined that the following adjacent queries contain at least one term in common (with the common term(s) stated in parentheticals): 1 and 2 ("palo alto"); 2 and 3 ("inns"); 7 and 8 ("san francisco"); and 8 and 9 ("inns").

The portions of these queries that are not in common are then identified as a candidate pair of equivalent descriptions for the same information need. (Stage 550). Continuing with the same example: "hotels" → "inns" is determined as a candidate pair from queries 1 and 2; "palo alto" → "san francisco" is determined as a candidate pair from queries 2 and 3; "hotels" → "inns" is determined as a candidate pair from queries 7 and 8; and "san francisco" → "palo alto" is determined as a candidate pair from queries 8 and 9.

As an alternative to using common terms to identify candidate pairs, the present invention may also be used to identify candidate pairs based on acronyms. This could be accomplished, for example, by comparing the letters of a term in one query with the first letters of terms in another query. For example, the query log in Fig. 6 shows a term "FDA" as query 5, and the terms "Food Drug Administration" as query 6. By comparing the letters of query 5 to the first letters of the terms of query 6, one might determine that "FDA"→"Food Drug Administration" is a candidate pair. Indeed, the terms in the subsequent query need not be adjacent, so that FDA could be mapped to "Food and Drug Administration".

Next, a score is calculated for each candidate pair. (Stage 560). In one implementation, the score is calculated as transform/A, where "transform" represents a candidate pair and "A" represents the total number of times the first half of the candidate pair occurs in the Data Set. Under this implementation, for example, the score for the candidate pair "palo alto" → "san francisco" would be 1 (the number of times the candidate pair appears in the Data Set) divided by 3

(the number of times "palo alto" appears in the Data Set), which yields a score of 0.333. Similarly, the score for the candidate pair "hotels → inns" would be 2/2, which yields a score of 1.0. In the foregoing examples, candidate pairs are treated as unidirectional (i.e., "palo alto" → "san francisco" is treated differently than "san francisco" → "palo alto"). Those skilled in the art will realize, however, that bi-directional candidate pairs may also be used consistent with the present invention.

The scores are then compared to a threshold value to determine whether the candidate pairs are to be treated as synonyms. (Stage 570). The threshold value can be defined in advance or in real time, depending on the confidence level desired. If the threshold is set quite high, for example, it is more likely that a candidate pair will represent synonyms if its score exceeds that threshold value. In one implementation, a threshold value of 0.1 yields suitable results, when used in combination with the scoring technique described above.

Fig. 7 is a flow diagram for identifying equivalent descriptions based on anchor text information, consistent with the present invention. For purposes of illustration, this process will be described in conjunction with the document system shown in Fig. 8, which shows three documents: A, B, C, and D. Document A contains anchor text "palo alto cars", which is hyperlinked to document C, and anchor text "palo alto inns", which is hyperlinked to document D. Document B contains anchor text "san francisco inns", which is also hyperlinked to document D. Document C presumably contains information about

palo alto cars, while document D presumably contains information about palo alto inns and san francisco inns.

The process begins by creating (or identifying) a list of anchor text units. (Stage 710). Using Fig. 8, for example, the list would consist of "palo alto cars", "palo alto inns", and "san francisco inns".

The anchor text units are then organized by the document being pointed to. (Stage 720). Accordingly, "palo alto cars" would be in one set, since it points to document C; and "palo alto inns" and "san francisco inns" would be in a second set, since they each point to document D.

Within each set, anchor text units containing one or more common terms are identified (stage 730) and the uncommon parts of those units are identified as a candidate pair (stage 740). Using Fig. 8, "palo alto inns" and "san francisco inns" are identified as containing common terms (i.e., "inns"), and "palo alto" → "san francisco" (the uncommon parts) are identified as a candidate pair.

A score is calculated for each candidate pair (stage 750) and the candidate pair is treated as synonyms if the score exceeds a threshold (stage 760). The score for each candidate pair may be calculated in a variety of ways, similar to those described above in reference to Fig. 5. For example, the score may be calculated as a ratio of the number of times the transform occurs divided by the total number of times the first half of the candidate pair occurs in the entire collection of anchor text units. Furthermore, the threshold may be set depending on the desired confidence level, also similar to that described above in reference to Fig. 5.

Finally, the present invention may be used not only to identify equivalent descriptions for an information need, but also alternative (or related) descriptions for an information need. For example, the invention may use search queries "hertz rentals" and "avis rentals" to determine that "hertz" and "avis" are equivalent (e.g., synonyms), whereas they may instead be alternatives. One way to exclude alternatives is to examine a set of information (the larger the better) to locate collections that contain one or both halves of the candidate pair. If both halves of the candidate pair occur frequently in such collections, one may deduce they are related or alternatives, rather than equivalents or synonyms.

For example, one might obtain a large set of documents that contain "hertz" and "avis". Within this set of documents, one would search for lists, tables, etc., that contain "hertz" and/or "avis". By then comparing the ratio of (1) the number of times both "hertz" and "avis" appear in a list or table, and (2) the number of times "hertz" appears in a list or table (or the number of times "avis" appears in a list or table), one can derive a score. This score can then be compared against a threshold to determine whether the halves of the candidate pair are alternatives or not.

C. Conclusion

The foregoing description of preferred embodiments of the present invention provides illustration and description, but is not intended to be exhaustive or to limit the invention to the precise form disclosed. Modifications

and variations are possible in light of the above teachings or may be acquired from practice of the invention.

The scope of the invention is defined by the claims and their equivalents.